PATENT **Docket No. RSW920030089US1**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS:	**David A. Selby**	Confirmation No. **1932**
APPLICATION NO.	**10/806,204**	
FILED:	**March 22, 2004**	Examiner: **A. Boyce**
CASE NO.	**RSW9200130089US1**	Group Art Unit: **3623**

TITLE: **SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR INCREASING THE EFFECTIVENESS OF CUSTOMER CONTACT STRATEGIES**

FILED ELECTRONICALLY ON December 19, 2008

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF DAVID A. SELBY UNDER 37 C.F.R. §1.131

Sir:

I, David A. Selby, hereby declares as follows:

1. I am the inventor of the subject matter claimed in the above-identified patent application.

2. I have been informed that U.S. Patent Application Publication No. 2005/0010472 to Quatse et al. ("Quatse") has been cited as prior art by the United States Patent and Trademark Office with respect to the above-identified patent application. I also have been informed that the filing date of Quatse is July 8, 2003.

3. This declaration is to establish conception and reduction to practice of the invention claimed in claims 1-24 of the above-identified application in the United States prior to July 8, 2003.

4. Prior to July 8, 2003, I conceived of and developed (i.e., reduced to practice) a working software program that met all of the limitations of claims 1-24 of my patent application, as more particularly detailed below.

5. Exhibit A attached hereto is an Invention Disclosure, prepared by me concerning the development of the invention disclosed and claimed in the above-identified patent application. The date appearing on the original of this document has been redacted from the copy attached hereto. However, the date is prior to July 8, 2003 and accurately reflects a date on which the document existed.

6. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code, and that such willful statements may jeopardize the validity of the application or any patent issued thereon.

5th Dec 2008
Dated

David A. Selby

Disclosure RSW8-2003-0180

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By David Selby On [redacted] 10:34:18 AM CET
Last Modified By David Selby On [redacted] 06:12:17 AM MST

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

***Title of disclosure (in English)**
A method for optimisation of marketing resources

Summary

Status		Under Evaluation
Final Deadline		
Final Deadline Reason		
*Processing Location		Raleigh - RSW
*Functional Area	select	(GS-Mike Haydoch/BICNS) GS-Mike Haydoch/BICNS
Attorney/Patent Professional		Gregory Doudnikoff/Raleigh/IBM
IDT Team	select	Joel R Grosh/San Diego/IBM
Submitted Date		[redacted] 10:49:24 AM CET
*Owning Division	select	GS
*Line of Business		BCS - Business Consulting Services Primary Inventor's Line of Business (LoB)
*Industry/Sector		Dist - Retail
*Competency		Consulting
Incentive Program		
Lab		
*Technology Code		687C
PVT Score		

Inventors with a Blue Pages entry

Inventors: David Selby/UK/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
Selby, David A (David)	086349	70/703211	7-246723	Pressley, Carl Nicholas

> denotes primary contact

Inventors without a Blue Pages entry

IDT Selection

Attorney/Patent Professional	Gregory Doudnikoff/Raleigh/IBM
IDT Team	Joel R Grosh/San Diego/IBM
Response Due to IP&L	[redacted]

REDACTED

REDACTED

REDACTED

REDACTED

To: David Selby/UK/IBM@IBMGB
cc: Carl Nicholas Pressley/UK/IBM@IBMGB
Subject: Submission of Disclosure Number RSW820030180
Security: ☒ IBM Confidential
Importance: ○ Urgent ○ Normal

DISCLOSURE NUMBER: RSW820030180
TITLE: A method for optimisation of marketing resources

REDACTED

To: Joel R Grosh/San Diego/IBM
cc: Gregory Doudnikoff/Raleigh/IBM
Subject: Evaluation of Invention Disclosure RSW820030180
Security: ☑ IBM Confidential
Importance: ○ Urgent ○ Normal

DISCLOSURE NUMBER: RSW820030180
TITLE: A method for optimisation of marketing resources

REDACTED

Disclosure: A method for optimisation of marketing resources

Introduction

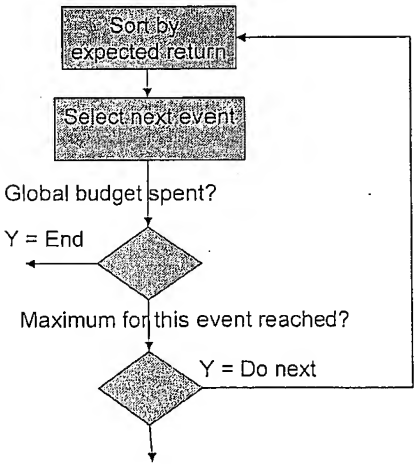
This disclosure describes a method for solving programmatically the business issue of optimising marketing resources. That is given a set of constraints of marketing materials, customers, budgets etc. An optimal stream of communication is selected which affords the "best" return on investment over time. This over approach has been disclosed in Docket Number P23,426 USA SYSTEM AND METHOD FOR INCREASING THE EFFECTIVENESS OF CUSTOMER CONTACT STRATEGIES

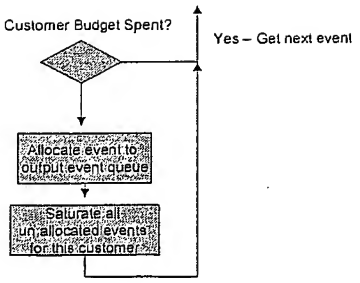
99-079
=
Bielli

This disclosure makes further claims, by describing a computer-based method, which optimal solves the problem described in the above, by utilising a novel extension to a well-known operational technique called a greedy heuristic. (See www.nist.com for formal definitions)

The method

The objective of the process is to develop a per customer contact stream, that is from the total list of marketing events available to an individual we wish to select a subset which maximises profit for the minimal cost, taking into account a set of business constraints. We propose this can be achieved in the following way:





This approach adopts a well know technique, a greedy heuristic. Although, due to the circumstances involved in the data it is possible to omit the intermediate sort between each application of the saturation. Saturation in nearly all applications reduces the target scores; the knowledge of this allows us to make a simple comparison of the next value to be selected with the subsequent. If this test fails then a sort is required, if this test passes then no sort is required. Omitting this continual sort significantly increases the performance of the technique and allows near real-time application of the approach.

Our Claims

We Claim an approach which:

1. Will solve efficiently enough to enable the solution to be applied to each individual, and therefore improve over the micro groups discussed in the referenced application.
2. By sorting only when required, the algorithm becomes highly scalable.
3. The algorithm achieves a solution that alleviates the need for linear programming.
4. The algorithm can be applied in a real-time application, to recompute the best contact strategy each time the underlying data is changed by customer interaction.

1. Due to the way the problem was being mathematically solved using a linear program in Bibelnieks it was necessary to solve for subsets of customers only on an irregular basis. As it would take large amounts of batch time to solve for a reasonably large portfolio of customers.
    a. The other implication of the Bibelnieks method was that you budget on a per Micro asset group basis. Which can be construed to be a very unrobust approach.

The improvement I have been able to make is to using a greedy Heuristic approach I have been able to solve for each individual customer, and therefore I can budget to an individual customer. Although I would argue the approach I use deviates from the written work enough to be unique, see (2)

2. The GH on its own would not necessarily give you a sufficiently large enough of a kick to allow to solve for an individual, but on examing the typical data used for this type of problem I have discovered a further optimisation.
3. I think Bibelnieks, please verify in case I missed talks about solving the problem once and using for a longish time. My appoach resolves the problem once per day for a constrained time window, say 30-90 days. So it is continually upto date. If you consider before Bibelnieks you would (and most still do) be in a six weeks cycle from generating a list of candidates and the mail dropping. Let alone planning customers, which I think by at least implication would be a long term planning activity.
4. The reason it is desirable to solve it once per day is that you need to be able to dynamically change the customers contact stream as soon as possible. Say it panned out the stream ended up hitting you with campaigns that sold you one particular item, say a winter coat. Well if I do it the Bibelnieks way and run my plan for the next few months, then one week in you buy a winter coat, every campaign you get after that point will be a waste of my marketing budget. Worse it will probably lead you to believe that I'm also stupid because you know we both know that you have a winter coat. So if you can update the plan the day something happens it becomes much more relevant for all subsequent contacts (campaigns). And improves the quality of the ROI from the stream of campaigns.

The GH flow is as follows:

From Bibelnieks you can see an individuals record would consist of a vector of scores for their propensity to respond, with some for of canalization and saturation applied, by a set of matrices. (See Bibelnieks figs 7-11) So you get, each being scored vertically. Normally a set of these would be input to the linear program plus the constraints.

Cannibalization and Saturation are taught by Bibelnieks in Figures 6-11 we would use the same or similar methodology in this approach. Now we deviate, Bibelnieks groups these customers into Micro segments based on some financial return curve for the group. We continue to work at an individual level, and also we operate a per customer budget based on the historical returns, are the ratio of marketing spend vs the amount of net profit for that customer.

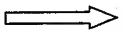
Customer No	C1	C2	C3	C4	C5	C6	C7
123456	0.2	0.3	0.565	0.9	0.32	0.53	0.56
123457	0.1	0.8	0.63	0.5	0.21	0.52	0.7

(a) Create a pivot table from the above record, which looks like this:
The important thing to note is the chronological order of the campaigns has not been destroyed. Campaign 1 occurs at T+0 and Campaign 2 occurs at T+1 and Campaign 3 occurs at T+2 etc etc.

Customer No	Campaign	Expected Gain
123456	1	0.2
123456	2	0.3
123456	3	0.565
123456	4	0.9
123456	5	0.32
123456	6	0.53
123456	7	0.56
123457	1	0.1
123457	2	0.8
123457	3	0.63
123457	4	0.5
123457	5	0.21
123457	6	0.52
123457	7	0.7

(b) Sort by expected Gain. (The implication of this step is that you are optimizing the list of contacts globally. (Ie: Independent of any one individual. Any event that creates the best expected gain will be top of the list)

Here a Radix sort is applied, The result is as follows:



Customer No	Campaign	Expected Gain
123456	4	0.9
123457	2	0.8
123457	7	0.7
123457	3	0.63
123456	3	0.565
123456	7	0.56
123456	6	0.53
123457	6	0.52
123457	4	0.5
123456	5	0.32
123456	2	0.3
123457	5	0.21
123456	1	0.2
123457	1	0.1

(fig 2)

Campaign	C1	C2	C3	C4	C5	C6	C7
Min Piece	1000.00	500.00	1000.00	1000.00	500.00	400.00	800.00
Max Piece	5000.00	1000.00	2000.00	3000.00	1500.00	1200.00	1500.00
Unit Price	0.62	0.56	0.43	1.03	0.93	1.20	1.04
Max Budg	3100.00	560.00	860.00	3090.00	1395.00	1440.00	1560.00
Min Budge	620.00	280.00	430.00	1030.00	465.00	480.00	832.00

(Apologies if this too small, its an imbed so you can blow up) (fig 3)

We entered the optimization procedure with some budgets (or otherwise known as constraints) the table fig 3 above shows for this run the proposed constraints. Number of pieces are decided typically by print run cut offs, that is the printing trade price their work based on economic cut offs, which factor the size of the printing equipment, cost of setup etc. Basically this normally means the more you print the cheaper it gets. In a call centre operation, these cut off's exist but are predicated by the number of operators capable of making outbound email calls, or having the right skill to introduce that type of campaign. (You may have 100 operators but only 10 are specialized in selling mortgage products for instance) For each you have a minimum and a maximum (Upper and lower bound) The cost is the monetary amount it takes to execute each campaign, this yield a maximum budget Pieces times cost. If you add together all of these budget values, then you have the maximum permitted spend for this run of the application. Typically you would want to stay under this

value, the objective is to generate the most return by spending the least. Simple example the goal is to make 100 dollars by spending the least, clearly if I can spend 30 cents and make a 100 its better than spending 50 dollars and making 100. Sure I'm stating the obvious but it's the underlying fundamental in this process. The global budget is set to between (sum min budgets and sum max budgets) This is also helpful because in business the typical thing that happens is that times get tough so every department needs to take say a 10% cut, to meet targets. It's easy to just specify 10% less overall in this process.

(c) Now If the global budget met, we stop. As we allocate a campaign to an individual we sum the number the amount spent. This gives us a comparison number.

(d) We may stop if there nothing that will yield a predicted profit. In the column Expected gain, fig 2 (the arrow moves downward but points to the next entry to consider. If this top entry is negative as we work down the list and continually apply saturation, ie: Once we've done something to an individual subsequent events become less appealing, so we reduce the expected gains.

(e) if we've got nothing left in the pool to allocate, we stop. Max pieces are hit across all events. (fig 3)

(f) Select top record and allocate that customer to that Campaign, before allocating, test if campaign has met budgetary constraint (total pieces or cost budget, you can see they are interchangable) This is done by maintaining counters for each, which are incremented after a successful allocation, also test to see if he's not exceeded this individual's budget. Maintain a counter per individual for their budget. If either is true, then simply discard the record and past to the next, by moving the arrow down one entry in fig 2.

(g) Delete that record from the global list.

(h) Locate that customer in the remaining records and apply saturation to all other campaign records that he has remaining unallocated. This achieved by using a set of indexes to the original data (Fig 1), so although for illustrative purposes Fig 2 has been re-arranged in the actual implementation we would have a vector of indexes. These are sorted and shuffled during the process. See Fig 4. Let's do the first completely. Sort index says take the fourth record as globally the best expected return. The fourth record is Campaign 4 for customer 123456. We set the campaign id to negative, to show it's been used. In Fig 4. We decrement the global count for campaign 4, and we deduct 1.03 from the global budget for customer (See fig 3) Now we apply saturation, which degrades the expected gain for campaigns 1-3 and 5-7. As we know this is the fourth, we step back to the first and iterate over each campaigns expected gain from 1-7 excluding any negative campaign id's. We have now applied the saturative effect of 4 vs 1, 4 vs 2, 4 vs 3, 4 vs 5, etc. Saturative effect is as per Bibelnieks Fig 11. and described in detail, we adopt the same approach for calculating this component. Due to this the expected gain will be reduced for this customer

for all other campaigns proceeding and following the allocated campaign.

Customer Number	Campaign id	Expected Gain
123456	1	0.1
123456	2	0.2
123456	3	0.4
123456	-4	0.9
123456	5	0.2
123456	6	0.53
123456	7	0.52
123457	1	0.1
123457	2	0.8
123457	3	0.63
123457	4	0.5
123457	5	0.21
123457	6	0.52
123457	7	0.7

Sort Index
4
9
14
10
3
7
6
13
11
5
2
12
1
8

Fig 4

(i) Now in a Standard GH you would totally resort the table. We don't. (That's compute intensive, think about it if you have 1m customers over 30 events, that's a 30m record sort, just how it expands...) It turns out that by the nature of saturation, you will suppress all the records remaining by some value, ie: If I give you this campaign it always makes it less likely that things in a near term time horizon will also be given. And we are solving on a say 90 day window, because we can do it that fast.
(j) Ok I've messed my flow a little, but bear with me, I should go to (c) but what I want to do is check the next event is greater that one after the event to be picked, if it isn't I swap the entries, and keep doing that until it is, the technique is similar to that employed in a bubble sort. (Air bubbles rising) Once you know the next is the best you simply continue at (c) Remember this works because you will only make all other events worse, lower value due to the nature of canbalisation/saturisation Look at Fig 5 this is pretty typical, the application of saturation has adjusted some items lower in the list but have had no effect on the next item to choose. (So traditionally you would sort blindly, in this algorithm you take a lazy evaluation approach) This wastes a significant number of computer cycles for no reason.

Customer Number	Campaign id	Expected Gain
123456	1	0.2
123456	2	0.3
123456	3	0.565
123456	4	0.9
123456	5	0.32
123456	6	0.53
123456	7	0.56
123457	1	0.1
123457	2	0.8
123457	3	0.63
123457	4	0.5
123457	5	0.21
123457	6	0.52
123457	7	0.7

Sort Index (Pass 1)	Sort Index (Pass 2)
4	4
9	9
14	14
10	10
3	6
7	7
6	13
13	11
11	3
5	12
2	2
12	5
1	1
8	8

Fig 5